Aprendendo linguagem de programação através da auto-explicação de exemplos em vídeo

Viviane Cristina Oliveira Aureliano[1,2]
Patricia Cabral Restelli de Azevedo Tedesco[2]

[1]Instituto Federal de Pernambuco – Campus Belo Jardim
Belo Jardim, PE – Brasil

[2] Universidade Federal de Pernambuco – Centro de Informática
Recife, PE – Brasil

{vcoa, pcart}@cin.ufpe.br

Abstract. *This paper presents an approach to learn programming through self-explanation of video recordings examples. In order to assess the effects of the proposed approach, we carried out an experiment with beginners in programming. Although significant differences were not found in the results, they are encouraging and they suggest further investigations are needed.*

Resumo. *O presente trabalho apresenta uma abordagem para o aprendizado de linguagem de programação a partir da auto-explicação de exemplos gravados em vídeo. Para avaliar os efeitos desta abordagem, realizamos um experimento com iniciantes nesta disciplina. Apesar de não terem sido encontradas diferenças significativas, os resultados obtidos são encorajadores e sugerem que novas investigações sejam realizadas.*

1. Introdução

Disciplinas introdutórias em linguagem de programação (LP) são intrinsecamente difíceis [ROBINS; ROUNTREE; ROUNTREE, 2003], comumente relacionadas aos seus altos índices de reprovação [BENNEDSEN; CASPERSEN, 2007] e, por estarem nos semestres iniciais de muitos cursos na área de tecnologia da informação, são possivelmente culpadas pelos altos índices de evasão nestes cursos no Brasil [SILVA FILHO et al., 2007]. De fato, os obstáculos para os estudantes nestas disciplinas são vários, sendo o maior deles não conseguir combinar e utilizar adequadamente os conceitos de programação aprendidos para a construção de um determinado programa [CASPERSEN; KÖLLING, 2009].

Para minimizar as dificuldades pelas quais os iniciantes em programação passam, lançamos mão de duas linhas principais de pesquisa que foram combinadas neste trabalho: (i) o framework *Stepwise Improvement* (FSI) [CASPERSEN, 2007; CASPERSEN; KÖLLING, 2009] e (ii) as auto-explicações (AEs) [CHI et al., 1989]. Utilizamos o FSI para estruturar e organizar os exemplos em vídeo utilizados como parte do material instrucional de um minicurso introdutório de programação. Aliado a isso, utilizamos questões para promover e orientar o processo de reflexão decorrente da auto-explicação dos participantes enquanto eles estudavam estes exemplos durante a execução do minicurso.

DOI: 10.5753/cbie.wcbie.2015.1359

Dentro deste contexto, o presente trabalho tem por objetivo apresentar uma abordagem de auto-explicação de exemplos em vídeo que unifica essas duas linhas de pesquisa. Ademais, descrevemos um experimento no qual analisamos os efeitos de aprender linguagem de programação através da abordagem proposta. Para alcançar estes objetivos, o restante deste trabalho encontra-se organizado da maneira descrita a seguir. Na Seção 2 apresentamos o seu referencial teórico. Na Seção 3 descrevemos a abordagem que unifica as linhas de pesquisa apresentadas na seção anterior. Na Seção 4 apresentamos o experimento realizado. Por fim, na Seção 5 apresentamos as conclusões e os trabalhos futuros decorrentes desta pesquisa.

2. Referencial teórico

Esta seção apresenta as linhas de pesquisa deste trabalho, FSI e AE, que são detalhadas nas próximas sub-seções.

2.1. Framework do *Stepwise Improvement* (FSI)

O FSI é um framework conceitual que descreve a atividade de programação como um processo sistemático e incremental que engloba três atividades diferentes, extensão, refinamento e restruturação [CASPERSEN, 2007; CASPERSEN; KÖLLING, 2009]. A atividade de extensão ocorre quando a especificação é estendida de maneira a cobrir mais casos de uso. A atividade de refinamento ocorre quando o código abstrato é modificado de maneira a se construir código executável que implemente a especificação correspondente. A atividade de restruturação ocorre quando é realizada uma melhoria dos aspectos não funcionais do programa, no entanto essa modificação não envolve uma mudança no comportamento aparente do programa. Estas atividades são organizadas em um espaço tridimensional que é percorrido pelos programadores enquanto eles estão construindo programas.

O FSI oferece uma contribuição significante à área de educação em programação fornecendo orientação nas três atividades que engloba, oferecendo orientação na maneira que materiais instrucionais (e.g., livro-texto, exercícios, aulas e exemplos) podem ser estruturados e na construção de programas propriamente dita. Neste sentido, o FSI foi empregado na estruturação do livro do *Greenfoot* [KÖLLING, 2010] e na definição de um processo para programação orientada a objetos [CASPERSEN, 2007].

2.2. Auto-explicação (AE)

Apesar do FSI fornecer orientação na estruturação e organização de materiais instrucionais, ele não oferece nenhum tipo de orientação na maneira que os alunos podem estudar e aprender programação a partir destes materiais instrucionais. Em virtude disso, propomos que este tipo de orientação aconteça através da AE [CHI et al., 1989]. A AE consiste em *"um diálogo mental que aprendizes possuem enquanto estudam um exemplo trabalhado e que os ajuda a entender o exemplo e a construir um esquema a partir deste exemplo"* [CLARK; NGUYEN; SWELLER, 2005]. De acordo com Chiu e Chi (2014), a atividade de auto-explicar, ou explicar para si mesmo, promove o aprendizado através da elaboração da informação que está sendo estudada, da associação dessa nova informação com a conhecimento prévio que o aprendiz possui, da construção de inferências e da conexão dos diferentes pedaços de informação.

A AE mostrou ser positiva para estudantes de variadas disciplinas, tais como biologia [CHI et al., 1994], física [CHI et al., 1989] e programação [BIELACZYC; PIROLLI; BROWN, 1995; VIHAVAINEN; MILLER; SETTLE, 2015], entre outras. Além disso, AEs podem ser obtidas a partir de diferentes formatos e modalidades de materiais instrucionais, incluindo exemplos em texto [BIELACZYC et al., 1995; CHI et al., 1989] e exercícios [VIHAVAINEN et al., 2015].

3. Auto-explicação de exemplos em vídeo

Os exemplos trabalhados[1] (*worked examples*) são uma sequência de passos que demonstram a realização de uma tarefa em particular ou a solução de um problema específico [CLARK et al., 2005]. De acordo com estes autores, os exemplos trabalhados são um tipo de material instrucional comprovadamente eficiente para alunos iniciantes em diferentes disciplinas, auxiliando-os na construção de novos esquemas enquanto eles estão adquirindo um novo conhecimento ou habilidade. Os exemplos trabalhados foram utilizados em diferentes disciplinas, tais como física [CHI et al., 1989], química [CRIPPEN; EARL, 2007] e programação [BIELACZYC et al., 1995; CASPERSEN, 2007], entre outros.

Os exemplos trabalhados são mais comumente apresentados de maneira estática, em formato textual [BIELACZYC et al., 1995; CHI et al., 1989; CRIPPEN; EARL, 2007]. Porém, eles também foram apresentados através de vídeos [CASPERSEN, 2007]. Os vídeos possuem o formato ideal para exibição de programas para iniciantes em uma disciplina de programação [BENNEDSEN; CASPERSEN, 2008]. Diferentemente dos livros que são estáticos e apresentam somente os programas resultantes, os vídeos têm a vantagem de poder combinar a apresentação dos programas com todo o seu processo de construção. Por esta razão, da mesma forma que Caspersen (2007), utilizaremos exemplos em vídeos estruturados de acordo com o FSI como parte do material instrucional disponível para os iniciantes em programação.

Como dissemos anteriormente, o FSI fornece orientação na organização dos exemplos em vídeos, no entanto, não fornece orientação na maneira como os alunos irão estudar estes exemplos. Por este motivo, durante a execução dos exemplos em vídeos, vamos estimular o processso de reflexão dos estudantes através da inclusão de questões para que eles auto-expliquem os exemplos que estão estudando. Espera-se com a inclusão destas questões que os estudantes sejam estimulados a refletir sobre os exemplos estudados e, consequentemente, aprendam a construir programas mais corretamente, aplicando as estruturas de programação aprendidas de maneira adequada do ponto de vista sintático e semântico.

Para que pudessemos incluir as questões nos exemplos em vídeo, estruturamo-os de forma que eles tivessem duas partes distintas. A primeira delas é a apresentação do enunciado do problema seguido por exemplos de execução do programa com entradas específicas e as saídas correspondentes. Em seguida, a segunda parte consiste na apresentação de uma possível sequência de passos que soluciona o problema apresentado. Para facilitar o processo de aprendizado dos iniciantes em programação, decidimos construir vídeos que apresentam somente sequências de atividades de

[1] Utilizaremos os termos *exemplos trabalhados* e *exemplos* indistintamente no decorrer deste trabalho.

extensões e refinamentos. Assim, a cada novo passo da solução executado no vídeo, teremos uma extensão e um refinamento.

As questões, por sua vez, foram adicionadas aos vídeos em quatro momentos distintos de sua execução: (i) um após a apresentação do enunciado do problema e de seus exemplos de entrada e saída; dois a cada novo passo da solução apresentada: (ii) um após se informar a funcionalidade que será implementada no passo e antes de se apresentar a implementação para o referido passo (quando uma extensão é executada) e (iii) outro após a implementação para o passo ser apresentada (quando um refinamento é executado); e (iv) o último após apresentar a solução completa do problema. As questões apresentadas no momento (i) tem por objetivo fazer com que o estudante auto-explique o propósito geral do programa apresentado no exemplo. As questões apresentadas no momento (ii) tem por objetivo fazer com os estudantes reflitam a respeito de uma possível solução para o passo de resolução do problema antes da solução do professor ser apresentada e, depois comparem a solução pensada por eles com a solução apresentada pelo professor. As questões apresentadas no momento (iii) tem por objetivo fazer com que os alunos façam inferências sobre a implementação apresentada pelo professor através da proposição de modificações no código apresentado. As questões apresentadas no momento (iv) tem por objetivo fazer com os alunos calculem mentalmente a saída do código a partir de uma determinada entrada, de forma que eles reflitam sobre a execução do programa apresentado.

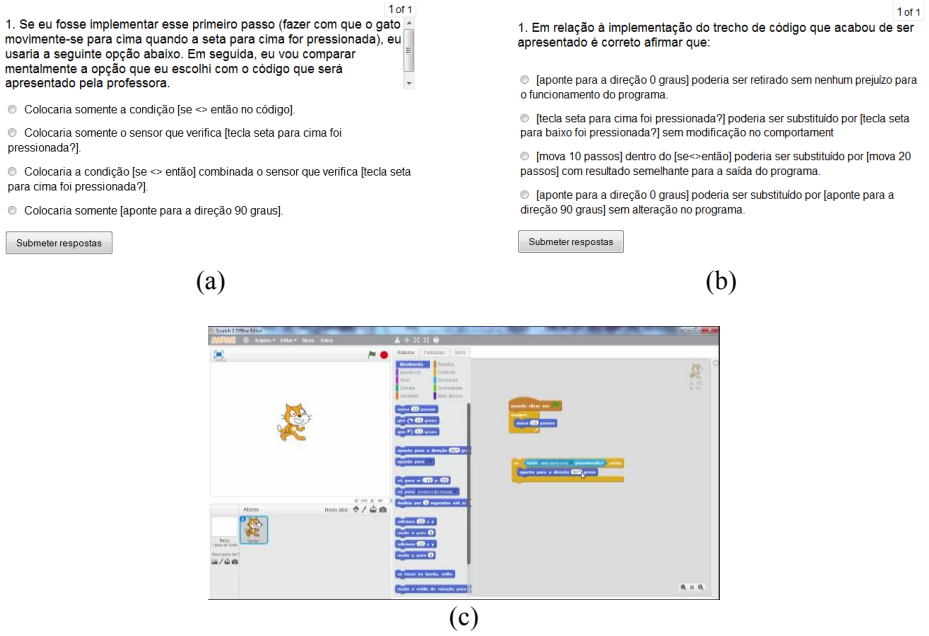


Figura 1. Trecho de exemplo em vídeo, apresentando passo da implementação do problema (c) juntamente com as questões anterior (a) e posterior (b) à implementação do passo do problema.

Para ilustrar esse procedimento, apresentamos na Figura 1 o trecho de um dos vídeos produzidos neste trabalho, onde utilizamos a ferramenta *Scratch* para a construção de um jogo. Este trecho apresenta o passo "fazer com que o gato movimente-se para cima quando a seta para cima for pressionada" que é parte do problema "fazer com que o gato se movimente através das quatro setas direcionais". Juntamente com o trecho de vídeo (Figura 1 (c)), apresentamos também as questões para os momentos

anterior (Figura 1 (a)) e posterior (Figura 1 (b)) à implementação do passo de resolução do problema. A cada questão apresentada, o vídeo pausa; em seguida, a questão é carregada e o estudante continua o vídeo após a submissão de sua resposta. Portanto, para que os períodos entre a parada do vídeo para responder as questões e a continuação da visualização do vídeo não fossem muito longos, todas as questões produzidas para os vídeos neste trabalho foram de múltipla escolha.

4. Descrição do experimento

O experimento realizado tem por objetivos: (i) analisar os efeitos da AE de exemplos em vídeo para o ensino introdutório de programação; e (ii) avaliar a qualidade e utilidade dos vídeos e das questões para AE do ponto de vista dos alunos. Sendo assim, as perguntas que norteiam a nossa pesquisa são as seguintes:

[QP1] O aprendizado de programação através da AE de exemplos em vídeo apresenta melhores resultados quando comparado ao aprendizado de programação somente através de exemplos em vídeo?

[QP2] Qual a opinião dos alunos em relação a qualidade e utilidade dos exemplos trabalhados em vídeo e das questões para a AE que foram utilizadas por eles?

4.1. Participantes

O experimento contou com 14 participantes voluntários. Estes participantes foram organizados em dois diferentes grupos, o grupo controle (GC) e o grupo instrucional (GI), de acordo com a disponibilidade relatada por eles para a participação no experimento. Durante a execução do experimento, o GC fez uso de exemplos trabalhados em vídeos, enquanto que o GI fez uso dos mesmos exemplos em vídeos em conjunto com as questões de AE desenvolvidas para estes vídeos.

4.2. Execução

O experimento foi realizado em uma instituição de ensino que atua na educação de nível médio, técnico e superior no Estado de Pernambuco, no primeiro semestre de 2015. Ele foi realizado em dois dias distintos (um para o GC e outro para o GI) e teve a duração de aproximadamente 4 horas. O experimento aconteceu no laboratório de informática da instituição de ensino e consistiu em uma sequência de 5 atividades: (i) questionário inicial; (ii) pré-teste; (iii) minicurso propriamente dito; (iv) pós-teste; e (v) questionário final. Antes da execução do experimento, a primeira autora do artigo explicou que o minicurso em questão se tratava de um experimento e coletou os termos de consentimento dos participantes (ou dos seus pais para aqueles menores de 18 anos). O questionário inicial teve por objetivo principal coletar informações demográficas dos alunos, além de informações sobre os seus hábitos de estudo e sua experiência prévia com vídeos para estudar. O pré-teste foi utilizado para avaliar o conhecimento prévio que os participantes possuiam antes do minicurso ser ministrado. O minicurso propriamente dito teve como principal objetivo de aprendizagem:

[OA1] Aplicar os conceitos de variáveis, estruturas de repetição e de decisão para a construção de um jogo usando a ferramenta *Scratch*.

Para alcançar este objetivo, o minicurso combinou períodos de aula expositiva seguidos por períodos de estudo individual dos exemplos em vídeo, conforme detalhado no Quadro 1. A aula expositiva foi ministrada e os exemplos em vídeo foram gravados pela primeira autora deste artigo. O pós-teste foi utilizado para avaliar o conhecimento adquirido pelos participantes logo depois do minicurso ser ministrado. Por fim, o questionário final teve por objetivo coletar a opinião dos participantes a respeito do material instrucional utilizado e do minicurso realizado.

Quadro 1. Detalhamento das atividades do minicurso.

Período	Conteúdo abordado
Aula expositiva	Introdução a programação com o *Scratch* Estrutura de repetição – **sempre**
Estudo dos exemplos em vídeo	Exemplo de aplicação da estrutura **sempre**
Aula expositiva	Estruturas de decisão – **se**
Estudo dos exemplos em vídeo	Exemplo de aplicação da estrutura **se**
Aula expositiva	Estruturas de decisão – **espere até**
Estudo dos exemplos em vídeo	Exemplo de aplicação da estrutura **espera até**
Aula expositiva	Variáveis
Estudo dos exemplos em vídeo	Exemplo de aplicação de variáveis

4.3. Material instrucional

Os materiais instrucionais produzidos para este experimento foram: (i) apresentação utilizada para guiar a aula expositiva; (ii) um conjunto de quatro vídeos que exibiam a aplicação dos conceitos apresentados na aula expositiva. Estes vídeos foram gravados pela primeira autora deste artigo; (iii) pré e pós-testes para avaliar o desempenho dos alunos antes e após o minicurso ministrado; (iv) um conjunto de perguntas para promover a AE no GI. Neste experimento, utilizamos o *Scratch* como linguagem de programação e ferramenta de desenvolvimento. Por esta razão, os jogos utilizados como parte do material instrucional desenvolvido foram baseados no *Curriculum Guide* do *Scratch* [BRENNAN; CHUNG; HAWSON, 2011].

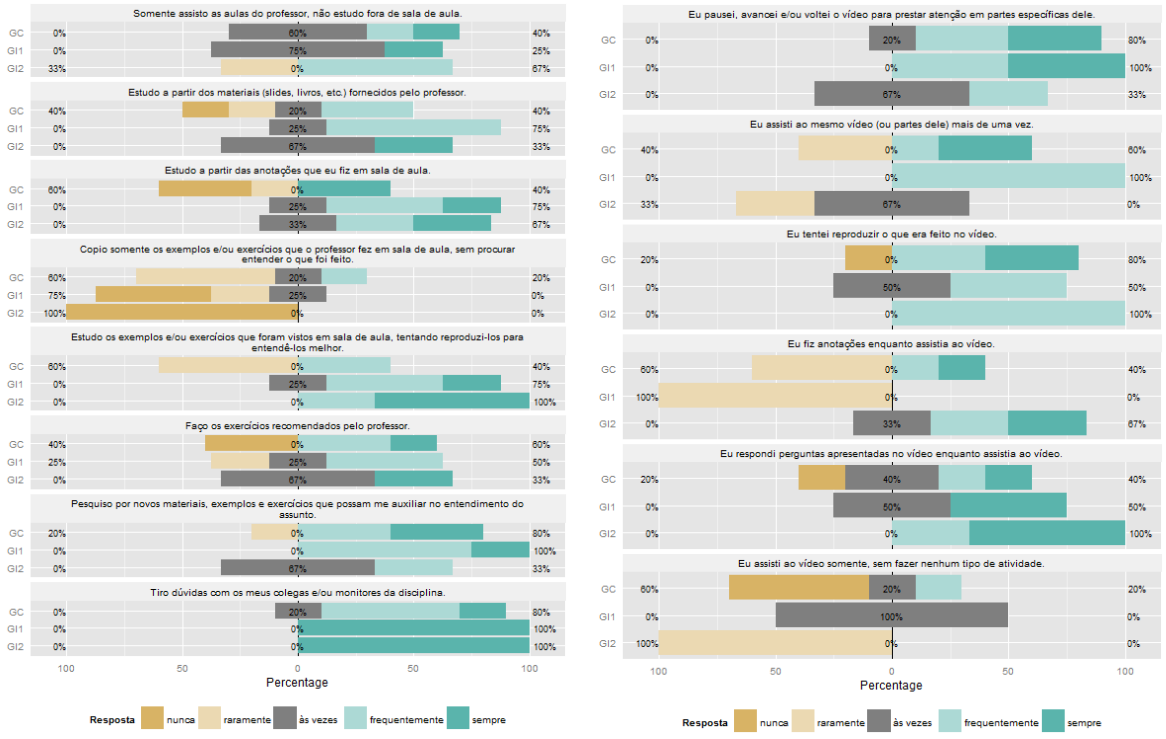
5. Resultados

Os resultados que obtivemos a partir dos dados coletados serão apresentados nas próximas sub-seções.

5.1. Perfil dos participantes

Dentre os 14 participantes, 11 haviam concluído o ensino médio e cursavam o ensino técnico profissionalizante e 3 cursavam o ensino médio integrado ao técnico profissionalizante. Além disso, os 11 participantes que concluiram o ensino médio eram todos da mesma turma e estavam cursando uma disciplina introdutória de LP em C. O fato da maioria dos participantes estarem cursando uma disciplina introdutória de LP não foi um impedimento à realização do experimento, visto que todos estes 11 participantes relataram sentir muitas dificuldades no conteúdo que foi ministrado até o momento da realização do experimento (o que foi comprovado posteriormente com o pré-teste realizado). Os outros 3 participantes, que ainda cursavam o ensino médio, não haviam cursado nenhuma disciplina relacionada à LP.

O GC contou com 6 participantes, no entanto, por problemas técnicos na coleta de dados de um deles, esta análise conta com os dados de somente 5 participantes. As idades deles variavam de 16 a 18 anos (média=17.2 anos). Todos eles já haviam concluído o ensino médio e estavam cursando o ensino técnico profissionalizante. Além disso, eles estavam cursando uma disciplina introdutória de LP em C, porém nenhum deles tinha experiência com o *Scratch*. Por estarem matriculados em uma disciplina de LP, os participantes foram perguntados sobre os seus hábitos de estudo para esta disciplina. As respostas obtidas são apresentadas na Figura 2 (a). Além disso, todos os participantes relataram ter utilizado vídeos disponíveis no Youtube para estudar para a disciplina de LP. Estes vídeos foram utilizados da maneira descrita na Figura 2 (b).



(a) (b)

Figura 2. Hábitos de estudo (a) e utilização de vídeos (b) pelos GC, GI1 e GI2.

O GI contou com 8 participantes, no entanto, por problemas técnicos na coleta de dados de um deles, os dados de somente 7 participantes puderam ser aproveitados nesta análise. As idades deles variavam de 15 a 19 anos (média=17 anos). Este grupo englobou participantes de dois perfis diferentes, o GI1 com participantes que já haviam concluído o ensino médio e estavam matriculados na disciplina introdutória de LP em C, e o GI2 com alunos que estavam cursando o ensino médio integrado ao ensino técnico e não haviam cursado nenhuma disciplina de LP. Nenhum dos participantes tinha experiência com o *Scratch*. Todos os participantes foram perguntados sobre os seus hábitos de estudo; os 4 participantes do GI1 responderam como estudavam para a disciplina de LP na qual estavam matriculados, enquanto os do GI2 responderam como estudavam para as disciplinas em geral. As respostas obtidas são apresentadas na Figura 2 (a). Além disso, 5 participantes relataram ter utilizado vídeos disponíveis no Youtube para estudar para a disciplina de LP (GI1) ou para estudar quaisquer outras disciplinas (GI2). Estes vídeos foram utilizados da maneira descrita na Figura 2 (b).

5.2. Desempenho dos participantes

Para responder a QP1, foram analisados os pré e pós-testes dos participantes de ambos os grupos. Os testes foram equivalentes e continham 7 questões contabilizando uma pontuação máxima de 10 pontos cada um. As primeiras 4 questões aplicavam os conceitos aprendidos em contextos semelhantes aos exemplos em vídeo estudados e, por isso, valiam 1 ponto cada uma. As 3 questões restantes aplicavam os conceitos aprendidos em contextos diversos àqueles apresentados nos exemplos e, por isso, valiam 2 pontos cada uma. Na correção dos testes, se o participante tivesse atingido o objetivo completo da questão utilizando as estruturas que seriam estudadas, ele ganhava a pontuação completa; se ele tivesse atingido o objetivo parcialmente utilizando quaisquer das estruturas disponíveis no *Scratch*, ele ganhava a metade da pontuação; e, por fim, o participante não ganhava nenhum ponto, se não conseguisse atingir o objetivo da questão. Os resultados obtidos por ambos os grupos no pré e pós-testes podem ser observados na Tabela 1 e na Tabela 2.

Tabela 1. Médias e desvios padrão (entre parênteses) dos resultados dos pré e pós-testes dos GC e GI.

	GC (n=5)	GI (n=7)
Pré-teste	0.30 (0.45)	0.29 (0.27)
Pós-teste	6.20 (3.49)	7.43 (2.76)

Analisando os dados apresentados na Tabela 1, podemos perceber que GC e GI não apresentam uma diferença estatisticamente significativa em relação aos resultados do pré-teste (U=16.50; z=-0.18; p=0.86) e, por isso, eles são comparáveis. Embora a média dos resultados do pós-teste do GI seja superior a média dos resultados do pós-teste do GC, os dados destes grupos também não apresentam uma diferença estatisticamente significativa (U=13; z=-0.77; p=0.44). Analisando os dados apresentados na Tabela 2, podemos perceber que GC e GI1 quando analisados em conjunto não apresentam uma diferença estatisticamente significativa em relação aos resultados do pré-teste (U=8.00; z=-0.59; p=0.56) e, por isso, eles são comparáveis. O mesmo acontece para GC e GI2 em relação ao pré-teste (U=4.50; z=-0.98; p=0.33) e, por isso, eles também são comparáveis. Analisando os resultados do pós-teste, verificamos que GC e GI1 não apresentam uma diferença estatisticamente significativa (U=6.00; z=-1.03; p=0.30). O mesmo acontece para GC e GI2 em relação aos resultados do pós-teste (U=7.00; z=-0.16; p=0.87).

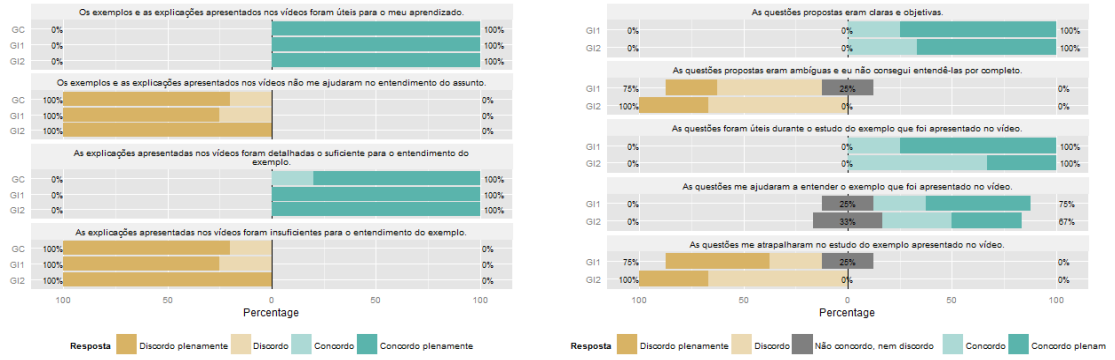
Tabela 2. Médias e desvios padrão (entre parênteses) dos resultados dos pré e pós-testes dos diferentes grupos instrucionais, GI1 e GI2.

	GC (n=5)	GI (n=7)	
		GI1 (n=4)	GI2 (n=3)
Pré-teste	0.30 (0.45)	0.13 (0.25)	0.5 (0.00)
Pós-teste	6.20 (3.49)	8.50 (1.91)	6.00 (3.46)

5.3. Opinião dos participantes

Para responder a QP2, analisamos as respostas dos participantes do experimento ao questionário final (adaptado de [RENKL, 2002]). Percebemos na Figura 3 (a) que os participantes de todos os grupos, GC, GI1 e GI2, tiveram uma opinião positiva a respeito da qualidade e utilidade dos exemplos e das explicações fornecidas pela professora nos vídeos. Percebemos também na Figura 3 (b) que a maioria dos

participantes do GI1 e GI2 teve uma opinião positiva a respeito da qualidade e utilidade das questões empregadas para promover a AE dos vídeos. As avaliações neutras também se fizeram presentes em algumas das respostas, porém elas foram em menor número.



(a) (b)

Figura 3. Opiniões dos GC, GI1 e GI2 a respeito dos exemplos em vídeo (a) e dos GI1 e GI2 a respeito das questões para AE nos exemplos em vídeo (b).

5.4. Discussão

Embora não tenha sido possível encontrar uma diferença estatisticamente significativa para os resultados obtidos no pós-teste, é interessante notar que, quando comparamos somente as suas médias, GI apresentou um resultado ligeiramente superior ao GC. Essa diferença entre os grupos é evidenciada quando analisamos os dados de GI separados pelo nível de instrução de seus participantes. Ao comparar GI1 com GC, cujos participantes possuem o mesmo nível de instrução, percebemos que a diferença das médias de GI e GC é superior a diferença das médias de GI1 e GC. Percebemos também que o desvio padrão do GI1 é inferior ao do GC, o que sugere que a aquisição de conhecimento por parte dos participantes do GI1 foi mais uniforme do que o dos participantes do GC.

Por outro lado, ao compararmos GI2 com GC, cujos participantes estão um nível de instrução acima dos participantes do primeiro grupo, percebemos que estes dois grupos apresentam desempenho e desvio padrão equivalentes, o que sugere que o conhecimento dos seus participantes no domínio de linguagem de programação foi nivelado com o minicurso. Desta maneira, esses resultados juntamente com as opiniões positivas dos participantes a respeito dos exemplos em vídeos e das questões para AE utilizados na nossa pesquisa são promissores e sugerem que novas investigações sejam realizadas.

6. Conclusões e trabalhos futuros

Neste trabalho, apresentamos uma abordagem para a AE de exemplos gravados em vídeo e também um experimento para analisar os efeitos da abordagem proposta. Apesar de não termos encontrado uma diferença estatisticamente significativa nos dados obtidos, quando comparamos as médias e os desvios padrão dos grupos encontramos que (i) GI teve um desempenho superior ao de GC; (ii) GI1 teve um desempenho superior e mais uniforme do que GC; e (iii) GI2 e GC tiveram um desempenho

equivalente, o que sugere que seus conhecimentos no domínio de programação foram nivelados. Além disso, tivemos uma avaliação positiva dos participantes no que diz respeito a qualidade e utilidade do material instrucional produzido. Juntos esses resultados são encorajadores para prosseguirmos com a nossa pesquisa que inclui o refinamento do material produzido e a realização de novos experimentos como trabalhos futuros.

Referências

BENNEDSEN, J.; CASPERSEN, M. Exposing the Programming Process. In: BENNEDSEN, J.;CASPERSEN, M.*, et al* (Ed.). **Reflections on the Teaching of Programming**: Springer Berlin Heidelberg, v.4821, 2008. cap. 2, p.6-16. (Lecture Notes in Computer Science). ISBN 978-3-540-77933-9.

BENNEDSEN, J.; CASPERSEN, M. E. Failure rates in introductory programming. **SIGCSE Bull.,** v. 39, n. 2, p. 32-36, 2007.

BIELACZYC, K.; PIROLLI, P. L.; BROWN, A. L. Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem Solving. **Cognition and Instruction,** v. 13, n. 2, p. 221-252, 1995/06/01 1995.

BRENNAN, K.; CHUNG, M.; HAWSON, J. Creative computing: A design-based introduction to computational thinking. **Retrieved May,** v. 9, p. 2012, 2011.

CASPERSEN, M. E. **Educating Novices in the Skills of Programming**. 2007. 323 DAIMI Aarhus University

CASPERSEN, M. E.; KÖLLING, M. STREAM: A First Programming Process. **Trans. Comput. Educ.,** v. 9, n. 1, p. 1-29, 2009.

CHI, M. T. H. et al. Self-explanations: How students study and use examples in learning to solve problems. **Cognitive Science,** v. 13, p. 145-182+, 1989.

CHI, M. T. H. et al. Eliciting Self-Explanations Improves Understanding. **Cognitive Science,** v. 18, n. 3, p. 439-477, 1994.

CLARK, R. C.; NGUYEN, F.; SWELLER, J. **Efficiency in Learning: Evidence-Based Guidelines to Manage Cognitive Load**. Wiley, 2005. ISBN 978-0-7879-7728-3.

CRIPPEN, K. J.; EARL, B. L. The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy. **Comput. Educ.,** v. 49, n. 3, p. 809-821, 2007.

KÖLLING, M. **Introduction to Programming with Greenfoot: Object-oriented Programming in Java with Games and Simulations**. Prentice Hall, 2010. ISBN 9780136037538.

RENKL, A. Worked-out examples: instructional explanations support learning by self-explanations. **Learning and Instruction,** v. 12, n. 5, p. 529-556, 10// 2002.

ROBINS, A. V.; ROUNTREE, J.; ROUNTREE, N. Learning and Teaching Programming: A Review and Discussion. **Computer Science Education,** v. 13, n. 2, p. 137-172, / 2003.

SILVA FILHO, R. L. L. E. et al. A evasão no ensino superior brasileiro. **Cadernos de Pesquisa,** v. 37, p. 641-659, 2007.

VIHAVAINEN, A.; MILLER, C. S.; SETTLE, A. **Benefits of Self-explanation in Introductory Programming**. Proceedings of the 46th ACM Technical Symposium on Computer Science Education. Kansas City, Missouri, USA: ACM**:** 284-289 p. 2015.